



# On line Mapping and Global Positioning for autonomous driving in urban environment based on Evidential SLAM

Guillaume Trehard, Evangeline Pollard, Benazouz Bradai, Fawzi Nashashibi

## ► To cite this version:

Guillaume Trehard, Evangeline Pollard, Benazouz Bradai, Fawzi Nashashibi. On line Mapping and Global Positioning for autonomous driving in urban environment based on Evidential SLAM. Intelligent Vehicles Symposium - IV 2015, Jun 2015, Seoul, South Korea. hal-01149504

**HAL Id: hal-01149504**

**<https://inria.hal.science/hal-01149504>**

Submitted on 7 May 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On line Mapping and Global Positioning for autonomous driving in urban environment based on Evidential SLAM

Guillaume Trehard<sup>1</sup>, Evangeline Pollard<sup>1</sup>, Benazouz Bradai<sup>2</sup> and Fawzi Nashashibi<sup>1</sup>

**Abstract**—Locate a vehicle in an urban environment remains a challenge for the autonomous driving community. By fusing information from a LIDAR, a Global Navigation by Satellite System (GNSS) and the vehicle odometry, this article proposes a solution based on evidential grids and a particle filter to map the static environment and simultaneously estimate the position in a global reference at a high rate and without any prior knowledge.

## I. INTRODUCTION

In the remaining challenges of autonomous driving in urban areas, vehicle accurate positioning in its environment is one of the most difficult tasks to tackle. A city offers indeed a large amount of situations in which some sensors are occluded (*e.g.* in urban canyons, traffic jams or crowded streets) or the road have been modified (*e.g.* road works, events) and at the same time, these situations require a high level of interpretation and knowledge to be correctly managed.

Current best answers to this problem are based on high resolution *data map* (*i.e.* pre-recorded map close to sensors representation) mostly coming from a 3D laser scanner [1] but also from cheaper 2D LIDAR [2]. Even with low-cost embedded sensors such as a camera [3], this map support indeed enables to reach a centimetre accuracy. However, these solutions could rapidly fail if their map is not updated quickly enough, if the road users are too numerous and occlude sensors or if the area is simply not mapped.

Another approach known as on-line *Simultaneous Localization And Mapping* (SLAM) [4], [5] enables to build a map of the crossed area while locating the vehicle in it. The results at a local scale can be very accurate but suffer a drift when considering long term driving or large maps.

On-line mapping supported by a GNSS information have then been proposed, mostly based on *landmarks maps* which are well adapted to visual SLAM [6], [7]. This GNSS support indeed leads to compensate the natural drift of SLAM solutions without any prior knowledge so that the estimate of both the pose and the surrounding map can be corrected with a GPS measure. If these *landmark maps* are interesting for the localization, their description of the environment is quite poor because limited to the landmarks themselves.

This information, without using high level 3D laser scan, can however be extracted from a basic 2D LIDAR sensor. A laser impact indeed provides both the information of the free

area crossed and about the impacted obstacle. Using such a sensor in urban environments becomes highly relevant when it comes to map the surroundings of a vehicle in terms of drivable areas and occupied zones.

If their mapping quality is not to be demonstrated any more, the *occupancy grids* mostly used as a discretized map of the surrounding area suffers from a drift which cannot be easily corrected and which lead to a growing error in the vehicle positioning. Moreover, this grid, applied to outdoor applications can lead to significant computation costs which are not suitable in an embedded system.

Based on the indoor localization solutions proposed in [8] and in [9], this article then introduces a solution for on-line mapping and global positioning using data from a 2D LIDAR sensor and a basic GNSS receiver. It enables to preserve a quasi-insignificant drift in a short range mapping and assure a consistent global positioning with a road level precision.

Combining the mechanisms of an Evidential SLAM [10] and of a Monte Carlo Localization close to the FastSLAM algorithm [8], this solution uses a strong approximation on the grid map construction which enables to scale the solution up to a vehicle size with a fast execution.

After a brief introduction to the Evidential SLAM concepts, this paper introduces the proposed Monte Carlo algorithm with the *common grid map assumption*. Some results and a discussion on the consistency of the filter are finally proposed in the last section.

## II. EVIDENTIAL SLAM

Using evidential theory in a Simultaneous Localization And Mapping algorithm has been proposed by the authors in [10] and validated in [11]. The contribution was to propose to switch from the classic probabilistic framework to the Transferable Belief Model (TBM) framework enables to bypass the static world assumption in most of the current SLAM processes. The developed algorithm was an adaptation of a Maximum-Likelihood SLAM (ML-SLAM) for evidential grids and with data coming from a LIDAR sensor. The output of the system was a 2D evidential grid which is used as the map of the environment and the position of the vehicle in this same map. An overview of the algorithm is proposed in Fig. 1 and the crucial points of the process are discussed in this section.

### A. Evidential grids

The main advantage of TBM over probabilities is to explicitly model the *not-known* and the *incoherent* information.

<sup>1</sup>RITS Team, INRIA, Domaine de Voluceau, 78150 Rocquencourt, FRANCE <https://team.inria.fr/rits>

<sup>2</sup>Valeo Driving Assistance Research France, 34 rue Saint Andre, 93012 BOBIGNY Cedex, FRANCE

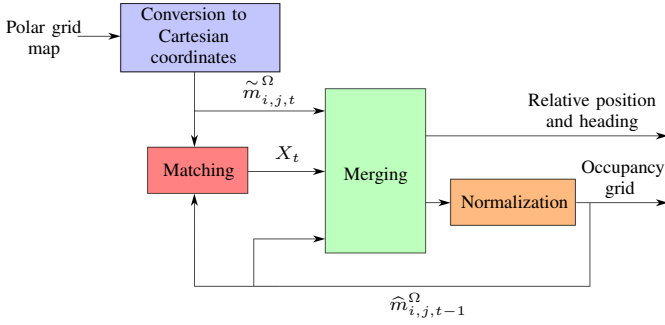


Fig. 1: Overview of the Evidential SLAM algorithm

Applied to occupancy grids, this property enables to describe the state of each cell of the grid with a set of four *masses*:  $\{Free, Occupied, Not-known, Conflict\}$  also denoted  $\{F, O, \Omega, \emptyset\}$ . This set is defined as the *Basic Belief Assignment* (BBA) of the cell and is updated through the time  $t$  with information coming from a LIDAR and according to the sensor model proposed by Moras *et al.* in [12].

This sensor model enhances both the area crossed by the laser beam and the laser impact itself. As illustrated in Fig. 2, a polar grid is used to model and discretize the laser scan. For each cell of the polar grid map, defined by its angle  $\theta$  and radius  $r$ , the measured BBA, denoted  $\tilde{m}_{r,\theta,t}^\Omega$ , is then filled as follows:

$$\begin{cases} \tilde{m}_{r,\theta,t}^\Omega(A) = \lambda \\ \tilde{m}_{r,\theta,t}^\Omega(\Omega) = 1 - \lambda \end{cases} \quad \text{with } A = \begin{cases} O & \text{if impacted} \\ F & \text{if crossed} \end{cases} \quad (1)$$

with  $\lambda$  the confidence accorded to the LIDAR sensor.

By converting those polar grids in a Cartesian reference and searching for the best match with the previously built evidential grid map, the SLAM process updates each cell status in the surrounding environment of the vehicle.

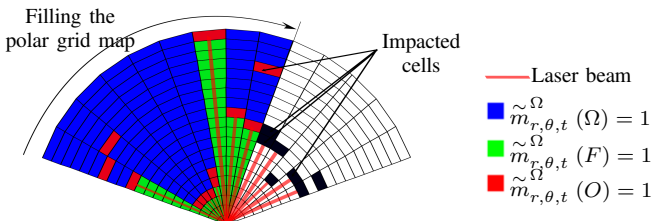


Fig. 2: Filling the polar grid map with a new laser scan

### B. Mapping quasi static environment

The *Conflict* represented in the TBM framework stands for the pieces of information which have been incoherent among different sources of information. In the current system, the evidential grid map is updated at each new laser scan so that the only source of information is the laser sensor. These incoherences can then occur in cases of differences between the measured states of a cell and the corresponding grid map cell. In an urban scenario and with a correct ego-localization,

the two events which can change a cell state detected by the LIDAR are the noise of the sensor itself (*i.e.* false alarms) or a moving obstacle, passing from one cell to another.

If the reliability of the laser sensor is good enough, one can assume that most of *Conflict* situations are the result of the mobile obstacles in the surrounding of the vehicle.

In its matching and merging operation, the proposed algorithm then balances the impact of these mobile objects in the SLAM process without any additive tracking system. The obtained evidential grid map is then assumed to represent the *quasi static* part of the environment. The adjective *quasi static* refers here to a period of time depending on the vehicle dynamics (*e.g.* High speed vehicles could see a large part of slow obstacles as static). In this article, terms *static* and *quasi static* will be used indistinctly to ease the read.

### C. Matching operator

The key part of the Evidential SLAM lays in its matching step. It aims at finding the optimum match between a new laser scan  $\tilde{M}_t$ , discretized as discussed in Sec. II-A, and the previously built evidential grid map  $\hat{M}_{t-1}$ .

The idea of the Evidential SLAM algorithm was to *propose* a set of candidates around an *a priori* displacement which corresponds to the possible matches between the polar grid (*i.e.* the new measure) and the evidential grid (*i.e.* the past measurement). Each candidate  $C$  is represented by a transformed version of the evidential grid  $\hat{M}_t^C$ . It is then scored to select the most likely candidate as the estimated vehicle displacement. In [10], the *a priori* displacement was computed with a basic *Constant Velocity* model and a discussion on the choice of the matching operator  $Op$  led to the following one:

$$Op(\hat{M}_{t-1}^C, \tilde{M}_t) = \sum_{\forall \text{cells}} f(\hat{m}_{i,j,t-1}^{\Omega,C}, \tilde{m}_{i,j,t}^\Omega) \quad (2)$$

where  $\hat{m}_{i,j,t-1}^{\Omega,C}$  is the BBA of a cell  $(i, j)$  in the occupancy grid reference and with the displacement corresponding to  $C$  and  $\tilde{m}_{i,j,t}^\Omega$  is the corresponding BBA in the polar grid map. This has the effect to sum scores of all the couples of cells from the measured polar grid map and the stored one. The function  $f$  is then defined:

$$f(\hat{m}_{i,j,t-1}^{\Omega,C}, \tilde{m}_{i,j,t}^\Omega) = \frac{(\hat{m}_{i,j,t-1}^{\Omega,C} \cup \tilde{m}_{i,j,t}^\Omega)(O)}{1 - (\hat{m}_{i,j,t-1}^{\Omega,C} \cap \tilde{m}_{i,j,t}^\Omega)(\emptyset)} \quad (3)$$

where  $\cup$  denote the disjunctive rule and  $\cap$  the conjunctive rule of the TBM [13].

This operator favours the cells with a BBA concentrated on the *Occupied* mass but balances their impact according to the conflict they create. *Conflict* situations (*i.e.* false alarms or mobile obstacles) will then be ignored or their impact will be limited in comparison to the static environment (*cf* Sec. II-B).

In the following sections of this article, another way to build a set of candidates will be proposed but the same operator  $Op$  will be used.

### III. SYSTEM ARCHITECTURE AND DETAILS

The system introduced in this publication aims at linking a local mapping provided by an Evidential SLAM (*cf.* Sec II) and a global positioning coming from a GNSS receiver. This fusion must be operated on-line so that a vehicle or a mobile robot can be located in real-time. As commented in [14], *Monte Carlo Localisation* (MCL) has several advantages to tackle the vehicle global positioning problem. MCL indeed performs well in non-linear and non-Gaussian situations, it does not require a complex initialization and it is particularly easy and fast to implement. Moreover, it eases the fusion of information from different kinds of sources and at different rates because its mathematical mechanism is more flexible with regards to different sensor models. This section introduces the theoretical background of the MCL and the architecture proposed to fuse global localization and local mapping.

#### A. Monte Carlo Localization

The MCL algorithm is based on Bayes filtering [9] and applied to mobile robot localization. It aims at estimating the *belief* denoted  $Bel$  (*i.e.* the posterior density) of a dynamical state  $x_t$  at time  $t$ , knowing all the past measurement data. Using the same notation as proposed in [14] the Bayes recursive filter theory then provides the following update equation:

$$Bel(x_t) = \eta p(o_t|x_t) \int p(x_t|x_{t-1}, a_{t-1}) Bel(x_{t-1}) dx_{t-1} \quad (4)$$

whith  $\eta$  a normalization factor,  $o_t$  the *observation data* and  $a_t$  the *action data* at time  $t$ .

In this equation, the densities  $p(o_t|x_t)$  and  $p(x_t|x_{t-1}, a_{t-1})$  are respectively known as the *perceptual model* and *motion model* which are both assumed as time invariant. Their notation are then simplified in [14] by  $p(o|x)$  and  $p(x'|x, a)$ .

The key idea of the MCL is to assume that a set of  $N$  weighted particles could sample the *belief*  $Bel(x_t)$ . Each particle is defined as a couple of a state  $x_t^i$  - a sample of  $x_t$  - and a weight  $w^i$ .

The algorithm is then divided in three main steps:

- A set of  $N$  particles is computed according to  $Bel(x_{t-1})$  approximated by the set  $\{x^i, w^i\}_{i=1\dots N, t-1}$ .
- A new set is then proposed by following the *motion model*  $p(x'|x, a)$  for each particle  $i$ .
- Each particle is then weighted according to its importance regarding the *perceptual model*  $p(o|x)$ .

The so formed new set of particles  $\{x^i, w^i\}_{i=1\dots N, t}$  represents the posterior density  $Bel(x_t)$ .

In practice, a resampling step is required when the set of particles is not efficient anymore to describe  $Bel(x_t)$ . To do so, the Sequential Importance Resampling algorithm [15] is used in the proposed system.

#### B. Evidential SLAM and MCL coupled

Since the proposed solution aims at estimating the global position of a mobile robot while building the map of its surrounding, the dynamical state  $x_t$  seen in Sec. III-A theoretically represents both the complete pose (position  $X$ ,  $Y$  and heading  $\Theta$  in Cartesian reference and displacement  $\Delta r$  and  $\Delta\Theta$  in current vehicle polar reference) of the robot itself and the evidential grid map of its surroundings  $M$ .

$$x_t = \begin{pmatrix} X \\ Y \\ \theta \\ \Delta r \\ \Delta\theta \\ M \end{pmatrix}_t \quad (5)$$

Applying the MCL algorithm to this dynamic state then leads to a set of  $N$  particles containing a sample  $x_t^i$  of  $Bel(x_t)$  and an importance factor  $w^i$ . It is important to notice that this sample then represents a realisation of both the vehicle pose and the grid map of its surrounding.

The *motion model* applied to each particle is chosen as the following non-linear evolution:

$$x_t^i = \begin{pmatrix} X^i \\ Y^i \\ \theta^i \\ a^i \\ M^i \end{pmatrix}_t \quad (6)$$

where  $a_t^i = \begin{pmatrix} \Delta r \\ \Delta\Theta \end{pmatrix}_t$ , the *action* sensor information, is a sample of the odometry measure density,  $M_t^i$  is the grid map transformed with the displacement represented by  $a_t^i$  and the evolution of the pose is proposed according to a *bicycle model*:

$$\begin{pmatrix} X \\ Y \\ \theta \end{pmatrix}_t^i = \begin{pmatrix} X \\ Y \\ \theta \end{pmatrix}_{t-1}^i + \begin{pmatrix} \Delta r_t \cos(\theta_{t-1} + \Delta\theta_t) \\ \Delta r_t \sin(\theta_{t-1} + \Delta\theta_t) \\ \Delta\theta_t \end{pmatrix}_t^i \quad (7)$$

Considering only the laser scanner as an *observation* sensor in this section, the *perceptual model*  $p(o|x)$  of the MCL (*cf.* Sec. III-A) can then be seen as the likelihood of the new laser scan knowing the *a priori* state, *i.e.* the *a priori* position and grid map corresponding to the vehicle at time  $t$ . The MCL is based on a set of samples so this perceptual model is equivalent to score each particle according to its state and the new laser scan. The proposition is then to use the matching operator seen in II-C to update the importance weight of each particle. This weight is updated following this equation:

$$w^i = Op(M_t^i, \tilde{M}_t) \quad (8)$$

A normalization step then occurs to assure that the sum of  $w^i$  over the  $N$  particles equals one.



### C. Common grid map assumption

If the MCL theory leads each particle to represent the pose of the vehicle along with the grid map of its environment, the amount of memory and computation power required to manage  $N$  grid maps can quickly increase and overpass reasonable resources. To bypass this limitation and obtain a fast algorithm, the proposition is to assume that the grid map obtained by merging each new scan according to the posterior estimation of the displacement  $\begin{pmatrix} \hat{\Delta r} \\ \hat{\Delta \Theta} \end{pmatrix}$  is a good approximation, for each particle, of its own grid map  $M_t^i$ . The same grid map can then be used to test each particle displacement using the same new laser scan:

$$M_t^i \simeq \hat{M}_{t|t-1}^i \quad (9)$$

where  $\hat{M}_{t|t-1}^i$  is the previous estimated grid map  $\hat{M}_{t-1}^i$  transformed with the displacement corresponding to particle  $i$ .

It enables to store a single grid map in memory which decreases the required computer resources. The importance weight of each particle is then updated using the following equation:

$$w^i = Op(\hat{M}_{t|t-1}^i, \tilde{M}_t) \quad (10)$$

If this can be seen as a strong assumption, results in Sec. IV show that the map itself is not more affected but only drifts as if there was no fusion. Since the algorithm aims at providing a correct map of the *direct* surrounding along with a global position, this drift is really soft in the concerned window so the common grid assumption softly affects the global positioning.

### D. GNSS updating

In addition to the laser scanner a GNSS receiver is assumed to be available as an *observation* sensor. The quality of this receiver is considered to be the same as a standard GPS. Its horizontal precision without any map-matching is then around 10 m and its rate is supposed to be 1 Hz. This rate is different from the laser one (*i.e.* approximately 10 times slower) so that it is assumed that a GNSS measure never occurs at the exact same time of a laser scan. Following the same process as in [14], the algorithm is then ran each time a new measure is coming, using either the laser scan or the GNSS measure. Considering this GNSS measure  $o_{Sat}$ , the *perceptual model* proposed in Sec. III-B is not adapted any more since GNSS information only contains a global position. Assuming that this position is affected by a white noise with a covariance of  $\sigma_{Sat}$ , proportional to the Horizontal Dilution Of Position (HDOP), the importance weight of each particle is then updated following this equation:

$$w^i \sim \mathcal{N}\left(\begin{pmatrix} X \\ Y \end{pmatrix}^i, o_{Sat}, \sigma_{Sat}\right) \quad (11)$$

*i.e.* the probability of the particle position  $\begin{pmatrix} X \\ Y \end{pmatrix}^i$  from the Gaussian distribution with the average  $o_{Sat}$ , the GNSS measure, and the standard deviation  $\sigma_{Sat}$ .

The Gaussian distribution used to represent the GNSS noise certainly appears as a strong simplification regarding other models but this system aims at highlighting the validity of the common grid map assumption (*cf.* Sec. III-C) via simulated GNSS measure. A more advanced model will be used in future works.

## IV. VALIDATION

### A. Results

In all the following results, the KITTI database is used as raw data input [16]. The following sensors are then simulated:

- A one layer, 360 deg LIDAR data is extracted from KITTI's Velodyne data.
- An odometry measure is created by adding a white noise with standard deviations  $\sigma_v = 0.3$  m/s and  $\sigma_w = 0.5$  rad/s to the true velocity and rotation speed.
- A GNSS measure is generated using the MATLAB GPS toolbox [17] with a standard deviation  $\sigma_{Sat} = 8$  m to simulate a classic GNSS signal.

As an example of result, the *Root Mean Squared Error* (RMSE) of the global position on a 2.2 km sequence of the KITTI database has been computed over 50 runs of the algorithm and plotted on Fig. 3. In addition, an example of an obtained result on another 2.5 km sequence can be found in Fig. 4.

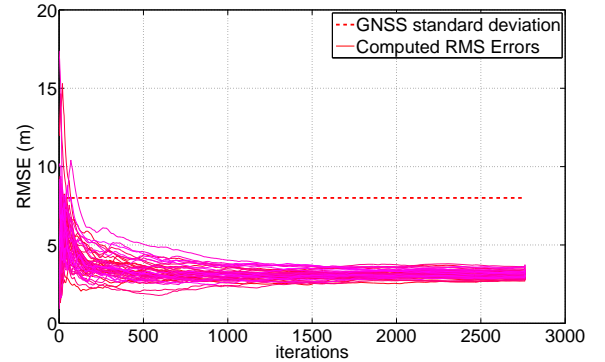


Fig. 3: RMS Errors of 50 runs of the algorithm on the same KITTI sequence

These results shows that the proposed filter converges to an RMSE approximately half of the standard deviation of the GNSS signal used. If this is still a 3 m error, it is worth noticing that it is achieved with low cost odometry and GNSS system supported by a reasonable cost LIDAR. Moreover, these results are obtained without any prior knowledge or map support so it shows the potential of this algorithm if those information were added.

In addition, the time of execution of the algorithm was between 60 ms and 70 ms (depending on the laser impact number) with 5000 particles and a local map of 100 m  $\times$  100 m

and a resolution of 0.2 m. The processor used was an Intel i5 without gpu support.

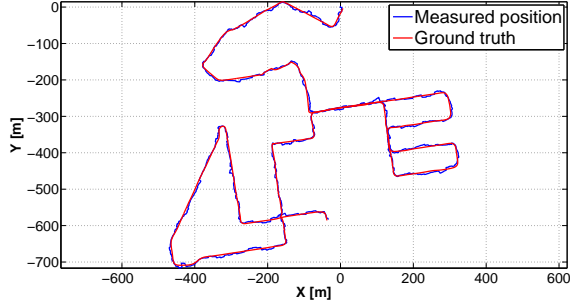


Fig. 4: Example of global positioning on a KITTI sequences

### B. Consistency

To measure the consistency of the proposed filter, the Normalized Estimation Error Squared (NEES) is used [18], [19]. It is used to check whether or not a filter can be considered as consistent by measuring, knowing the true state  $x_t$ , the NEES coefficient  $\epsilon_t$ :

$$\epsilon_t = (x_t - \hat{x}_t)P_t^{-1}(x_t - \hat{x}_t)^T \quad (12)$$

where  $P_t^{-1}$  denote the inverse covariance matrix associated to the estimated state  $\hat{x}_t$ .

This coefficient is averaged over  $M$  Monte Carlo runs of the algorithm. When  $M$  approaches the infinite, the proof of consistency is validated if  $\epsilon_t$  tends to the dimension of the considered state for each step  $t$  of the algorithm.

A simulation has then been run over  $M = 50$  Monte Carlo tests on the same sequence of the KITTI database as the one used to compute the RMSE (*cf.* Sec. IV-A). The results are plotted in Fig. 5.

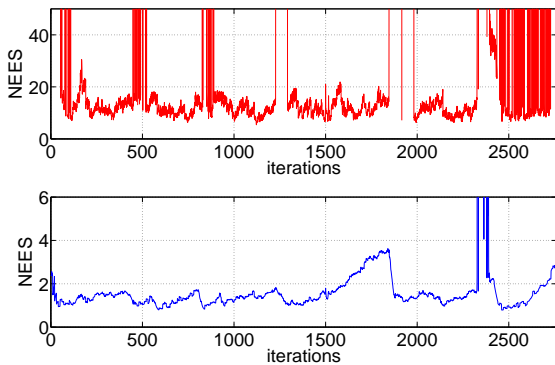


Fig. 5: up: NEES corresponding to the complete pose  
down: NEES with a restricted state to the global position and heading

The first plot refers to the NEES of the pose  $\begin{pmatrix} X \\ Y \\ \Theta \\ \Delta r \\ \Delta \Theta \end{pmatrix}$

introduced in Sec. III-B. It shows fast augmentations of the NEES coefficient largely over its supposed limit (*i.e.* the state dimension: 5 here) which signifies that the filter is optimistic. This results might be linked with the one obtained by T. Bailey *et al.* in [18]. A particle filter is indeed used to perform a SLAM algorithm and the optimistic character of the filter is linked to the resampling step which leads to loose track of some displacement hypotheses of the SLAM itself. The common grid map approximation explained in Sec. III-C is moreover strongly inconsistent so that the displacement related part of the state might suffers this approximation too.

The second plot confirms these hypotheses. The NEES refers there to the *global* position part of the state  $\begin{pmatrix} X \\ Y \\ \Theta \end{pmatrix}$  and the plot shows a much more reasonable result of a filter slightly conservative.

It enables to conclude that the global positioning of the proposed filter is a safe approximation of the true position but that the drift which affects the mapping part remains.

### C. SLAM drift

As explained in Sec III-C, a common grid map is used for each particle of the MCL algorithm to approximate the correct map. This hypothesis is assumed to be valid since only the *direct* surrounding of the environment is considered and the corresponding grid map then not suffer a strong drift. To validate this approximation, the algorithm was tested over 11 different sequences of the KITTI database and the drift affecting the grid map was computed according to the method proposed in [16].

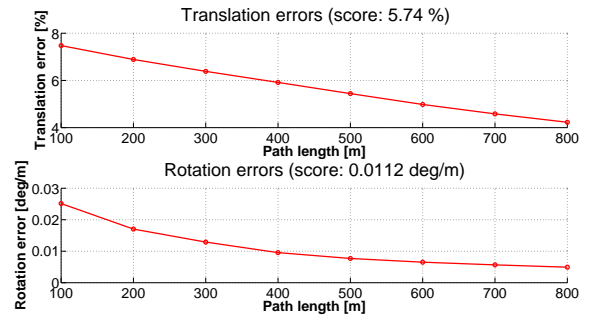


Fig. 6: Drift of the map on 11 sequences of the KITTI database

The results plotted on Fig. 6 show the drift which affects the map and illustrate that it is under 5 % in displacement and 0,03 deg/m in rotation. Those results are similar to the ones obtained with the Evidential SLAM in terms of local positioning [11], [10]. An example of the tested series is plotted Fig. 7.

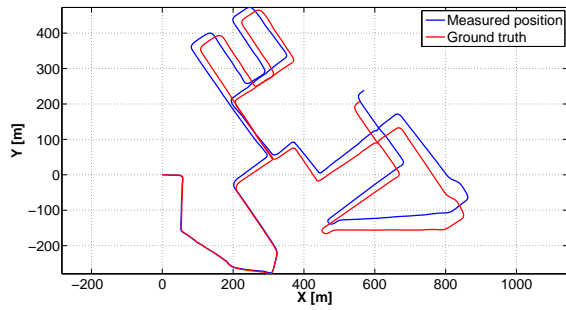


Fig. 7: Example of local positioning on a KITTI sequence

The common grid map assumption can then be validated as an interesting alternative for on-line mapping of the *direct* environments of the vehicle.

#### D. Discussion

One can notice that the SLAM algorithm could be seen as an odometry measure in the proposed system. Through the *observation model*  $p(o|x)$  seen in Sec. III-B, the SLAM process indeed balances the displacement of each particle. As a consequence, the SLAM score infers on the global process only *via* the estimation of the vehicle displacement, exactly like an odometry.

On another hand, the GNSS signal only provides information about the global positioning part of the algorithm. Its impact on the global process is then restricted to the the global pose on the vehicle.

This way of fusing a GNSS and a local mapping then enables to provide a good local navigation while searching for its coordinates in global references. If these global coordinates are only at a road level of accuracy, the measured displacement is accurate around the decimetre so can support a planning or control system.

In addition, the global navigation quality is sufficient to be matched with a map database so can provide a direct link between a local and a global description of the scene.

#### V. CONCLUSION

A solution to fuse a GNSS localization with an Evidential SLAM using a particle filter have been proposed and tested in this article. This solution enables to both locate a vehicle in a global reference and map its surrounding. The map of the surrounding suffers a drift which can be ignored when considering only the direct direct environment so that it can be used for control purpose. The filter is slightly conservative so that the output position provided is a safe estimation which can then be used in a map-matching algorithm. If the global performances still have to be improved, this system is a fast solution which can easily be implemented on an embedded system. It finally opens lots of perspectives for urban driving such as on-line mapping using both the information of a global map database and of the embedded sensors.

#### REFERENCES

- [1] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," pp. 4372–4378, 2010.
- [2] J. Xie, F. Nashashibi, M. Parent, and O. G. Favrot, "A real-time robust global localization for autonomous mobile robots in large environments," *11th International Conference on Control Automation Robotics & Vision (ICARCV 2010)*, pp. 1397–1402, 2010.
- [3] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving," pp. 176–183, 2014.
- [4] H. F. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.
- [5] T. Bailey and H. F. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 108–117, 2006.
- [6] J. Carlson, "Mapping Large, Urban Environments with GPS-Aided SLAM," Ph.D. dissertation, Carnegie Mellon University, 2010.
- [7] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," 2002.
- [8] D. H. D. Fox, W. Burgard, and S. Thrun, "A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
- [9] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," pp. 1322–1328 vol.2, 1999.
- [10] G. Trehard, Z. Alsayed, E. Pollard, B. Bradai, and F. Nashashibi, "Credibilist Simultaneous Localization and Mapping with a LIDAR," in *International Conference on Intelligent Robots and Systems (IROS)*, 2014.
- [11] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, "Credibilist SLAM Performances with Different Laser Set-ups," in *ICARCV 2014-13th International Conference on Control, Automation, Robotics and Vision*, 2014.
- [12] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," *IEEE International Conference on Robotics and Automation (ICRA 2011)*, pp. 84–89, 2011.
- [13] G. Shafer, *A mathematical theory of evidence*. Princeton university press Princeton, 1976, vol. 1.
- [14] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 12, pp. 99–141, May 2001.
- [15] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," pp. 107–113, 1993.
- [16] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pp. 3354–3361, 2012.
- [17] A. K. Tetewsky and A. Soltz, "COLUMNS-INNOVATION: GPS MATLAB TOOLBOX REVIEW," *GPS World*, vol. 9, no. 10, pp. 50–57, 1998.
- [18] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," pp. 424–429, 2006.
- [19] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM Algorithm," pp. 3562–3568, 2006.